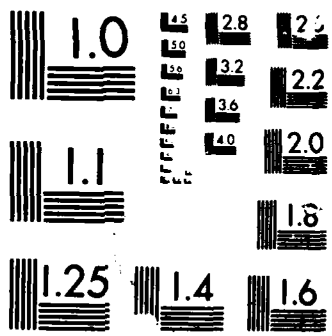
AD-A176 163 A GENERAL FRAMEWORK FOR MODELING PRODUCTION(U) 1/1
CALIFORNIA UNIV BERKELEY OPERATIONS RESEARCH CENTER
S T HACKMAN ET AL JUL 86 ORC-86-9 N00014-76-C-0134

UNCLASSIFIED F/G 13/8 NL


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.8
3.2
3.6
4.0

OPERATIONS RESEARCH CENTER

A GENERAL FRAMEWORK FOR MODELING PRODUCTION*

by

Steven T. Hackman[1]
Robert C. Leachman[2]

ORC 86-9 July 1986

UNIVERSITY OF CALIFORNIA

BERKELEY

12

A GENERAL FRAMEWORK FOR MODELING PRODUCTION*

by

Steven T. Hackman[1]
Robert C. Leachman[2]

ORC 86-9 July 1986

*Sponsored in part by a gift to the University of California from Intel Corporation and also by the Office of Naval Research under Contract N00014-76-C-0134 with the University of California. Reproduction in whole or in part is permitted for any purpose of the United States Government.

[1]School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332.

[2]Department of Industrial Engineering & Operations Research, University of California, Berkeley, California 94720.

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ORC 86-9	2. GOVT ACCESSION NO. ADA176163	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A General Framework for Modeling Production		5. TYPE OF REPORT & PERIOD COVERED Technical
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Steven T. Hackman Robert C. Leachman		8. CONTRACT OR GRANT NUMBER(s) N00014-76-C-0134
9. PERFORMING ORGANIZATION NAME AND ADDRESS Operations Research Center University of California Berkeley, CA 94720		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR 337 015
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Department of the Navy Arlington, VA 22297		12. REPORT DATE July 1986
		13. NUMBER OF PAGES 46
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Production Planning, Model Formulation, Linear Programming, MRP, CPM

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

See Report

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

A GENERAL FRAMEWORK FOR MODELING PRODUCTION

Steven T. Hackman

School of Industrial and Systems Engineering
Georgia Institute of Technology

Robert C. Leachman

Dept. of Industrial Engineering and Operations Research
University of California, Berkeley

ABSTRACT

~~We~~ introduce a *general framework* for developing and evaluating deterministic models of production processes. We evaluate familiar linear programming planning formulations, Manufacturing Resources Planning (MRP) and Critical Path Methods (CPM) in terms of the framework. Flaws are exposed in linear programming formulations incorporating lead times. We ~~provide~~ a correct reformulation, as well as a more general formulation accommodating non-integer lead times and unequal-length planning periods. Redefining MRP and CPM illustrates the framework's use for comparing seemingly unrelated models and for rigorously explaining limitations in existing models. ~~We propose~~ a structured approach for the development of accurate models of production. proposed

1. INTRODUCTION

Researchers in production planning and scheduling are beginning to realize that their work to date has seen a disappointing level of use in industrial practice. For example, Abraham *et al.* [1985] point out "a disparity between the insights, orientation and methodologies of past [production] research and the possible scientific bases of current practice." They stress the need for a "unifying framework" to help focus, position and evaluate future research.

Although most researchers and practitioners would agree that a unifying framework is lacking, there is little consensus as to what it should consist of or what needs it should address. Consider, for example, the following proposed frameworks. Abraham *et al.* [1985] suggest a hierarchical framework of decision levels corresponding to a presumed hierarchy of decision-making. Dempster *et al.* [1981] propose a general framework for the analytical evaluation of hierarchical planning systems. Bensoussan *et al.* [1985] offer a "unified mathematical treatment of production planning and production smoothing problems, in the framework of optimal control theory." Akinc and Roodman [1986] propose a new framework for modeling aggregate production planning problems in which emphasis is placed on flexibility in specification of the decision options and the relevant cost structure. Geoffrion [1985] proposes a "structured modeling" framework to facilitate the data-base development for application of pre-specified operations research models.

Surely, representation of the appropriate decision alternatives, decision hierarchy, and cost structure, as well as the effort required in data-base development, are important concerns for planning and scheduling. However, there is another, equally fundamental concern which attracts our research efforts. Embedded in each planning and scheduling model is a *model of production* - a mathematical representation of the set of technologically feasible operations or actions within the production process. While the body of literature proposing optimal or near-optimal algorithms for production planning and scheduling models is voluminous, a very narrow range of models of production is considered in this literature, as well as in the above-cited frameworks. Clearly, an accurate representation of the production process is critical to the optimality and even feasibility of generated plans and schedules. Therefore, we suggest that a framework is needed to (1) assess the accuracy or validity of the representation of a particular production process by a given production model, and (2) guide the development of more accurate production models when existing models provide inadequate representations of the production process.

In this paper, we introduce a *general framework* for developing and evaluating deterministic models of production. The framework incorporates familiar modeling principles such as flow conservation and activity analysis, but it expresses these principles in terms of new abstract *model elements*. When existing models of production are expressed in terms of the framework, extensions and generalizations become apparent. Moreover, the mathematical language and structure of the framework provides a foundation to compare and contrast seemingly unrelated models arising in such diverse environments as discrete-parts manufacturing, continuous-flow production, and project management.

The framework is introduced in Section 2. In the sequel, we redefine in terms of the framework three of the most familiar production planning models: standard linear programming formulations, Manufacturing Resources Planning (MRP), and Critical Path Methods (CPM). Each model highlights different abstract model elements of the framework, and serves to illustrate different uses of the framework.

In section 3, we use the framework to develop more accurate models of production. Familiar linear programming (l.p.) planning models are analyzed, including those which incorporate "production lead times". We consider physical phenomena which give rise to lead time and show that the physical phenomena should be categorized into three distinct types. Flaws in familiar formulations are exposed, and we derive a correct reformulation from the framework. We then generalize the formulation to accommodate non-integer lead times and unequal-length planning periods.

Section 4 illustrates use of the framework for explaining limitations of existing models. We show that MRP techniques are based on an incomplete model of production. As a consequence, a simple explanation is provided of why MRP techniques will in general lead to excessive work-in-process inventories.

Section 5 illustrates the value of a coherent language for comparing production model elements. We interpret CPM as a model of production. Recognizing that a strict precedence relationship between two activities is a form of inventory balance, it becomes clear how CPM can be usefully extended by incorporating activity analysis of l.p. models.

Section 6 concludes the paper with a brief outline of how the framework can be used in a structured modeling approach for the development of accurate production models.

2. A GENERAL FRAMEWORK

Our development of a general framework for models of production follows from research in production theory initiated by R.W. Shephard. Shephard [1970] developed the first axiomatic steady-state model of production. The model elements in his framework are production correspondences which abstractly model the input-output possibilities of the technology. The first general activity network model was developed in Shephard et al. [1977], which introduces intermediate product transfers between activities and activity dynamic production correspondences. Hackman [1984] extends this framework, replacing activity correspondences with activity dynamic production functions. Our presentation is based in part on Chapter 2 of Hackman [1984].

The framework is a *meta-mode'* ...ction. It delineates precisely what is and what is not a valid production model in much the same way that the "meta-model":

$$\begin{aligned} \min\ & c \cdot x \\ & A \cdot x = b \\ & x \geq 0 \end{aligned}$$

delineates what is and what is not a linear program. The framework identifies the abstract model elements, or building blocks, and the relationships between the elements, which are required in a mathematical representation of a production system. The development of a deterministic model of a production system is viewed as a specification of the elements of the framework so as to reflect the particular technologies and processes of the system. We now define these elements.

The Activity Network

The framework models a production system as a directed network whose nodes represent *primitive production activities*, i.e., activities whose internal organization is not further modeled. Directed arcs indicate possible transfers of *intermediate products*. Intermediate products are outputs of one or more activities which serve as inputs to other activities. Production at each activity requires intermediate products transferred from other activities and/or *system exogenous inputs.* System exogenous inputs are of two types: non-storable services and storable materials. Services include labor trades, machines and facilities, while materials include purchased parts, raw materials, fuels, etc. An exogenous material may be the same as an intermediate product of the system.

Allocations of system exogenous inputs are represented as transfers from an initial node A_0. In a system with N activities, transfers of final products are represented as transfers to a sink node A_{N+1}. A final product may be the same as an intermediate product of the system, as in the case of spare parts. Thus, a production system is regarded as a jointly operating, finite number of interrelated primitive production activities $A_1, A_2, \ldots, A_N$ which use system exogenous inputs of goods and services to produce final outputs.

We emphasize that the activity network is merely an abstract representation of work flow and need not represent any physical arrangement of facilities.

Modeling the Flows of Goods and Services

Since flows of goods and services (inputs and outputs) have a dynamic character, we model each flow as a bounded function of time, where time is modeled by the nonnegative part of the real line. Time 0 is a reference point defining the point in time after which flows would be determined by planning calculations using the model.

There are two fundamental types of flows. In the first and more common type, called a *rate-based flow*, $x(\tau)$ represents the rate-- quantity per unit time-- of flow at time τ. In the second type, called an *event-based flow*, $x(\tau)$ is a numerical representation of an event at time τ. For example, an event-based flow in a project-oriented production system might be

$$x(\tau) = \begin{cases} 1 & \text{if } \tau \text{ is the project completion time} \\ 0 & \text{otherwise.} \end{cases}$$

In another example, suitable for batch transfers of intermediate products, $x(\tau)$ might indicate the quantity transferred at time τ. Unless specifically identified as event-based, all flows in this paper are rate-based.

For each model, an index set $\Lambda = \{t_k\}_1^\infty$ of epochs is defined. The t_k's are ordered so that $0 = t_0 < t_1 < \cdots$. Each $I_k = (t_{k-1}, t_k]$ is a *period*, the points in Λ are called *time-grid points*, and Λ is called the *time grid*. Both lower and upper bounds on the lengths of the periods are assumed. We note that the existence of a finite time grid is usually assumed in discrete-time planning and control. For event-based flows included in the model, the set Λ indicates the *possible* times of events.

It will be convenient when representing inventory balance equations to work with *cumulative* flows. For a flow x, the corresponding cumulative flow is denoted by the upper case letter X, and is defined by $X(t) = \int_0^t x(\tau)d\tau,\ t \geq 0$. Conversely, the unintegrated flow corresponding to the cumulative flow X is denoted by the lower case letter x.

Unless otherwise stated, if a component of a vector of flows is not explicitly defined, then it is assumed to be the zero function.

Intermediate Product Transfers

We index the different inputs and outputs of the system from 1 to K with the first M indices denoting products or materials and the last $K - M$ indices denoting non-storable services such as labor and machine time. Let $v_{ij} = (v_{ij}^1, v_{ij}^2, \ldots, v_{ij}^M)$ denote the vector of flows representing transfers of products from activity i, $i = 0,1,\ldots,N$, sent to activity j, $j = 1,2,\ldots,N+1$. These transfers may not be immediately received at activity j. To distinguish transfers sent from those received, the framework associates with v_{ij} the vector $\hat{v}_{ij} = (\hat{v}_{ij}^1, \ldots, \hat{v}_{ij}^M)$ of transfers from activity i received at activity j. For example, if there is a constant time lag l_{ij} for shipments from i to j, then

$$\hat{v}_{ij}(t) = \begin{cases} v_{ij}(t - l_{ij}) & \text{if } t > l_{ij} \\ \text{prespecified} & \text{otherwise.} \end{cases}$$

We remark that the transformation between transfers sent and transfers received is not modeled as a production activity of the network. If such a transformation requires scarce resources such as material handling equipment, then the activity network should be re-defined to represent the material handling as an activity.

Allocations, Transfers and Applications

The framework further distinguishes transfers received at an activity from inputs actually applied in production. Let $w_i = (w_i^{M+1}, w_i^{M+2}, \ldots, w_i^K)$ denote the allocation flows of system exogenous services to activity i and let $y_i = (y_i^1, y_i^2, \ldots, y_i^K)$ denote the flows of inputs actually applied in production at activity i.

In each model, a joint domain D for the allocations, applications and transfers ($\{w_i\}$, $\{y_i\}$ and $\{v_{ij}\}$) reflects the domain of technologically feasible flows. The domain may involve complicated constraints linking applications at different activities. For example, suppose activities A and B represent the production of products A and B, respectively. Suppose each activity uses the service of the same machine which can only process one type of product at a time. Then the domain for applications must be defined to ensure that applications of the machine service at activities A and B are not simultaneous.

Dynamic Production Functions

For each activity i, a dynamic production function f_i maps input flows $y_i = (y_i^1, \ldots, y_i^K)$ applied at activity i into realized output flows $f_i(y_i) = (f_i^1(y_i), \ldots, f_i^M(y_i))$. The domain of the production function f_i (as derived from the joint domain D) is denoted by D_i. Typically, one or more components of y_i will be the zero function. Note that we depart from the traditional economic definition of a production function in two respects. First, our production functions map *applications* into *realized outputs* rather than *allocations* into *possible outputs*. Second, our production functions do not relate steady-state rates of inputs and outputs, but rather they map functions of time representing technologically feasible input applications into functions of time representing realized outputs.

The Model of Production

Given an identification of the model elements and their domains, the framework specifies a deterministic model of production in terms of flow conservation constraints as follows. Let $I_i^m(t)$ denote the inventory at time t at activity i of product or material m, $1 \le m \le M$. At each time $t \ge 0$, the cumulative supply of product or material m at activity i - including production, received transfers and initial stock - is

$$I_i^m(0) + F_i^m(Y_i)(t) + \sum_{j=0}^{N} \hat{V}_{ji}^m(t). \tag{2.1}$$

and the cumulative amount of product or material m transferred from and used by activity i is

$$\sum_{j=1}^{N+1} V_{ij}^m(t) + Y_i^m(t). \tag{2.2}$$

For each product or material m, inventory balance constraints require that for each activity i and all time $t \ge 0$,

$$I_i^m(t) = I_i^m(0) + F_i^m(Y_i)(t) + \sum_{j=0}^{N} \hat{V}_{ji}^m(t) - \sum_{j=1}^{N+1} V_{ij}^m(t) - Y_i^m(t) \ge 0. \tag{2.3}$$

We refer to (2.3) as the *fundamental inventory balance equation*. The framework models inventories of products at each activity allowing for completed products awaiting transfer and transferred products awaiting application. Hence, a particular product may have multiple inventory locations. It is important to note that material in process, either in transit between activities or inside an activity production process, is not considered to be inventory.

On the other hand, for each non-storable input m, $M+1 \le m \le K$, we require for each activity i and all time $t \ge 0$ that

$$y_i^m(t) \le w_i^m(t). \tag{2.4}$$

Equation (2.4) expresses the conservation of service m by activity i.

Let $u = (u^1, u^2, \ldots, u^M)$ denote a vector of final output flows. For each product m and all time $t \ge 0$, inventory balance constraints at the sink require that

$$I_{N+1}^m(0) + \sum_{i=1}^{N} \hat{V}_{i,N+1}^m(t) - U^m(t) \ge 0. \tag{2.5}$$

Finally, the framework models conservation of exogenous inputs as follows. Let $c^m(t)$ denote the system supply or capacity of exogenous input m at time t. If input m is storable, then the cumulative allocation may not exceed the cumulative supply, i.e., for all time $t \geq 0$,

$$\sum_{i=1}^{N} V_{0,i}^{m}(t) \leq C^m(t). \tag{2.6}$$

If input m is non-storable, then for all time $t \geq 0$,

$$\sum_{i=1}^{N} w_i^m(t) \leq c^m(t). \tag{2.7}$$

In the case of dynamic processes, activity output flows and transfers-received flows from time 0 until some time after time 0 could be functions of applications and transfers-sent which occur at or before time 0. We assume that all flows which are consequences of applications or transfers-sent at or before time 0 are fixed and prespecified in planning calculations.

From the perspective of the general framework, a deterministic model of a specific production process is categorized and explicitly described by identifying and defining the abstract model elements: the activity network, the exogenous inputs, the intermediate and final products, the activity dynamic production functions, and the domains for the allocations, applications and transfers. While the activity network, inputs and products are well-defined in most presentations of the models we consider in follow-on sections, the production functions and the domains are typically only implicitly treated. We will thus focus our attention on these model elements.

Rather than repeatedly delineate assumptions common to various models, it is convenient to define the following model *categories*. In an *acyclic* production model, the activity network is acyclic. In a *product-generated* production model, each activity produces a single product not produced by any other activity. In a *discrete-time* production model, each rate-based flow is a step-function constant in each time period. In addition, a finite-horizon is assumed. In a *normal* production model, (1) activities do not receive products they can not apply, and (2) activities do not transfer products they can not produce.

3. LINEAR PROGRAMMING MODELS

We now illustrate applications of the framework for developing more accurate models of production. Starting with explicit sets of assumptions about the physical process, we use the framework to derive linear programming models of production. We show that familiar formulations incorporating time lags do not correctly model the physical phenomena they purport to represent. We provide a correct formulation, and we also generalize the formulation to accommodate non-integer lead times and unequal length time periods, cases which have not been treated before.

3.1. Leontief Production Functions

Most familiar linear programming models of production employ a very simple class of activity production functions. In particular, if the applications of inputs are positive, then they must be *proportional* and therefore may be indexed in terms of one profile. The outputs produced are also assumed proportional and indexed by the same profile which indexes the inputs. We refer to the profile as the *intensity* of the activity.

Casting this in the language of the framework, we say that the domain D_i of a production function f_i is *Leontief* if there are non-negative constants a_i^m, $m = 1,2, \ldots, K$ such that

$$D_i \subset \{ y_i \mid y_i^m(t) = a_i^m z_i(t),\ m = 1,2, \ldots, K \text{ for some intensity curve } z_i \}$$

As each curve y_i^m in a Leontief domain D_i may be expressed in terms of its defining intensity curve z_i, we write $f_i(z_i)$ instead of $f_i(y_i)$. We say that a production function f_i with Leontief domain is itself Leontief if there are non-negative constants c_i^m, $m = 1,2, \ldots, M$ such that $f_i^m(z_i) = c_i^m z_i$, $m = 1,2, \ldots, M$. (The production functions are so named due to their similarity to the steady-state functions used in Leontief [1951].) The coefficient a_i^m represents the rate input m is applied to A_i per unit intensity, and the coefficient c_i^m represents the rate product m is produced by A_i per unit intensity.

If an activity's production process is modeled by a Leontief production function, then the model assumes production is instantaneous and time-invariant. That is, an output rate of an activity at a point in time is solely a function of input application rates at the same point in time; moreover, this input/output transformation is the same at all points in time. If output at time t is actually a function of input applications over some period of time, then the Leontief function is inappropriate.

3.2. Basic Linear Programming Models Without Time Lags

The following assumptions about the physical system lead to a linear programming model of production:

(1) The system is represented by a product-generated (normal) activity network.

(2) Each activity dynamic production function is Leontief and the only constraints on its domain D_i *are those implied by the Leontief assumption. (This characterizes the Dynamic Linear Activity Analysis Model (DLAAM)* presented in Shephard *et al.* [1977].) As a result of assumption (1), activity intensity will be measured in units of output. Thus, $f_i^i(z_i) = z_i$, $i = 1,2,\ldots,M$.

(3) There are no domain constraints on allocations or transfers (other than non-negativity).

(4) A product is held in inventory only at the activity which produces it.

(5) Intermediate product transfers are "instantaneous", i.e., transfers sent equal transfers received. Combining this assumption with assumptions (3) and (4), as intermediate products are transferred, they are applied. In the language of the framework,

$$v_{ij}^i = \hat{v}_{ij}^i = y_j^i = a_j^i z_j .$$

(6) Transfers of final product are instantaneous. In view of assumptions (3) and (4), transfers of final product are equated to final output requirements, i.e., $v_{i,N+1}^i = u^i$.

(7) The only exogenous inputs are the non-storable services, i.e., $v_{0i}^m = 0$, $m = 1,\ldots,M$, $i = 1,\ldots,N$.

(8) Since domain constraints on allocations are assumed to be trivial, allocations are equated to application, i.e., $w_i^k = y_i^k$, $k = M+1,\ldots,K$.

From assumptions (1)-(8), we obtain the following model of production: For each intermediate product i, inventory balance equation (2.3) reduces to

$$I_i^i(t) = I_i^i(0) + Z_i(t) - \sum_{j=1}^{M} a_j^i Z_j(t) - U^i(t) \geq 0, \text{ all } t \geq 0. \tag{3.1}$$

For each non-storable service k, the allocation constraint (2.7) becomes

$$\sum_{i=1}^{N} a_i^k z_i(t) \leq c^k(t), \text{ all } t \geq 0. \tag{3.2}$$

To develop a linear programming model, we invoke the additional assumptions of a discrete-time production model with unit length time periods (0,1], (1,2], $\cdots$. (As is customary, we refer to the interval $(t-1,t]$ as "period t".) Note that on each unit interval, all flows in and out of inventory of product i are at constant rates; hence the rate of change of the inventory level during each interval is constant. If we enforce nonnegativity of the inventory level of product i at the end points of an interval, it follows that the inventory must be nonnegative during the entire interval. Hence, inventory balance can be guaranteed for all $t \geq 0$ simply by enforcing (3.1) at the time grid points.

Equations (3.1) and (3.2) now may be rewritten as, respectively,

$$I_i^i(t) = I_i^i(0) + \sum_{\tau=1}^{t} z_{i\tau} - \sum_{\tau=1}^{t} \{ \sum_{j=1}^{M} a_j^i z_{j\tau} + u_\tau^i \} \geq 0, \ i = 1,2, \ldots, M, \ t = 1,2, \cdots, \tag{3.3}$$

$$\sum_{i=1}^{M} a_i^k z_{it} \leq c_{kt}, \ k = M+1, \ldots, K, \ t = 1,2, \cdots, \tag{3.4}$$

where $z_{i\tau}$, u_τ^i, c_{kt} denote constant rates of flows during the respective time periods. In more familiar form, (3.3) and (3.4) are expressed as

$$I^i(t) - I^i(t-1) = x_{it} - \sum_{j \neq i} a_j^i x_{jt} - u_t^i, \ i = 1,2, \ldots, M, \ t = 1,2, \cdots, \tag{3.5}$$

$$I^i(t) \geq 0, \ i = 1,2, \ldots, M, \ t = 1,2, \cdots, \tag{3.6}$$

$$\sum_{i=1}^{M} a_i^k x_{it} \leq c_{kt}, \ k = M+1, \ldots, K, \ t = 1,2, \cdots, \tag{3.7}$$

where x_{it} denotes the amount produced of product i in period t, $I^i(t)$ denotes the inventory of product i at time t, u_t^i denotes the final demand for product i in period t, a_i^k denotes the amount of service k required per unit of product i, and a_j^i denotes the amount of product i input per unit output of product j. We have just shown that physical assumptions (1)-(8) lead to the constraints (3.5)-(3.7) in the traditional linear programming multi-period multi-stage aggregate production planning model (see, for example, Hax and Candea [1985] or Johnson and Montgomery [1974]).

The assumptions under which (3.5)-(3.7) comprise a valid model of production are now clear. If the transformations of activity input to output are not proportional, not time-invariant, or not instantaneous, or if transfers become significant either because there are shipment lags or because transfers are not applied immediately (e.g., inspections), then (3.5)-(3.7) is no longer a valid model of production. (We remark that assumptions (2), (3), (5)-(8) are necessarily assumed when one posits the model (3.5)-(3.7); however, alternatives to (1) and (4) exist which also lead to (3.5)-(3.7).)

3.3. Linear Programming Models With Time Lags

One particularly unrealistic aspect of the assumptions leading to the previous linear programming model is the requirement that processes occur instantaneously. For example, assumption (2) insists that output from an activity depends only on input to that activity at the same instant in time. This assumption precludes operations with significant post-processing lags (e.g., time for steel to cool), and also operations with significant processing times (e.g., metal-cutting). Likewise, assumption (5) precludes situations with significant transfer times.

Considerable research has been devoted to developing practical linear programming models without these instantaneousness assumptions. In all of this work, all significant time lags are lumped under the single heading "lead-time". For example, Billington *et al.* [1986] describe lead time as a "nonproduction lag, such as the time for paint to dry, hot metal to cool, or a batch to be physically moved between production areas." They develop a linear programming model in which the inventory balance constraints are of the form:

$$I^i(t) = I^i(0) + \sum_{\tau=1}^{t} z_{i(\tau - L_i)} - \sum_{j=1}^{M}\sum_{\tau=1}^{t} a_j^i z_{j\tau} - \sum_{\tau=1}^{t} u_\tau^i \geq 0, \; t = 1, 2, \cdots . \tag{3.8}$$

Here, L_i, the "unavoidable lead time" for product i, is defined so that output of product i started in period t may first become available in period $t + L_i$. Inherent in the definition of L_i is the assumption that each unavoidable lead time is independent of the production rate and can be expressed as an integral number of time periods.

We now demonstrate that these parameters can not correctly model the stated phenomena. To do so, we define three different types of lags contributing to lead time and formulate a correct linear programming model assuming these time lags are integral.

Output Lag

In some production processes, a product can not be transferred or released to inventory for a period of time after it is "produced". For example after painting, a fixed amount of time may be required for the paint to dry; or after lumber is sawed, a fixed amount of time may be required to inspect and grade the output. For such processes, assumption (2) must be modified as follows: Input applications at A_i are still proportional, but the output curve is shifted by a constant amount $LA_i \geq 0$. (The symbol 'LA' denotes "lag after" application). In the language of the framework,

$$f_i^i(z_i)(\tau) = \begin{cases} z_i(\tau - LA_i) & \text{if } \tau > LA_i \\ \text{prespecified} & \text{for } 0 < \tau \leq LA_i . \end{cases}$$

Note that applications of inputs are still instantaneous, but consequent output emerges after a time lag LA_i. Note also that, since applications are prespecified before time 0, output of product i must be prespecified for $\tau \in (0, LA_i]$. Substituting

$$F_i^i(Z_i)(t) = \int_0^t f_i^i(z_i)(\tau) d\tau = \int_{-LA_i}^{t - LA_i} z_i(\tau) d\tau,$$

(2.3) becomes

$$I_i^i(t) = I_i^i(0) + \int_{-LA_i}^{t - LA_i} z_i(\tau) d\tau - \sum_{j=1}^{M} a_j^i Z_j(t) - U^i(t) \geq 0, \; \text{all } t \geq 0. \tag{3.9}$$

Transfer Lag

A period of time may be required to transfer parts between activities. Suppose a transfer from activity i to activity j requires exactly LT_{ij} time units. (The symbol 'LT' denotes "lag-transfer".) Retaining our other assumptions, assumption (5) must be modified as follows:

$$y_j^i(\tau) = a_j^i z_j(\tau) = \hat{v}_{ij}^i(\tau) = \begin{cases} v_{ij}^i(\tau - LT_{ij}) & \text{if } \tau > LT_{ij} \\ \text{prespecified} & \text{for } 0 < \tau \le LT_{ij}. \end{cases}$$

Note that $z_j(\tau)$ must be prespecified for $\tau \in (0, LT_{ij}]$ because it corresponds to transfers sent from A_i at or before time 0. Retaining the output lags and noting that

$$V_{ij}^i(t) = \int_{LT_{ij}}^{t+LT_{ij}} \hat{v}_{ij}^i(\tau) d\tau,$$

(2.3) now becomes

$$I_i^i(t) = I_i^i(0) + \int_{-LA_i}^{t-LA_i} z_i(\tau) d\tau - \sum_{j=1}^{M} \int_{LT_{ij}}^{t+LT_{ij}} a_j^i z_j(\tau) d\tau - U^i(t) \ge 0, \text{ all } t \ge 0. \tag{3.10}$$

The intensity of each A_k must now be prespecified for $-LA_k < \tau \le \max_i \{LT_{ik}\}$.

Input Lag

Another type of lag arises when there is a delay between the time material is received at an activity and the time it is applied. For example, there may be an inspection of inputs required which does not utilize scarce resources. Alternatively, such a lag may be used to approximate queue time required because of cyclic batch production on a single machine, or other complex domain constraints. We further modify assumption (5) so that there is a lag of exactly $LB_{ij} \ge 0$ time units between transfer receipt at A_j of intermediate product i and its application at A_j. (The symbol 'LB' denotes "lag before" application.) In the language of the framework the modified assumption (5) is

$$y_j^i(\tau) = a_j^i z_j(\tau) = \begin{cases} \hat{v}_{ij}^i(\tau - LB_{ij}) = v_{ij}^i(\tau - LT_{ij} - LB_{ij}) & \text{if } \tau > LT_{ij} + LB_{ij} \\ \text{prespecified} & \text{for } 0 < \tau \le LT_{ij} + LB_{ij}. \end{cases}$$

Retaining our other modifications and observing that

$$V_{ij}^{i}(t) = \int_{LT_{ij}}^{t+LT_{ij}} \hat{v}_{ij}^{i}(\tau)d\tau = \int_{LT_{ij}+LB_{ij}}^{t+LT_{ij}+LB_{ij}} a_j^i z_j(\tau)d\tau,$$

(2.3) now becomes

$$I_i^i(t) = I_i^i(0) + \int_{-LA_i}^{t-LA_i} z_i(\tau)d\tau - \sum_{j=1}^{M} \int_{LT_{ij}+LB_{ij}}^{t+LT_{ij}+LB_{ij}} a_j^i z_j(\tau)d\tau - U^i(t) \geq 0, \text{ all } t \geq 0. \quad (3.11)$$

With the additional lag, the intensity of each A_k must now be prespecified for $-LA_k < \tau \leq \max_i \{ LT_{ik} + LB_{ik} \}$.

It is important to note that the fixed lag LB_{ij} between transfer receipt and application predetermines a quantity $a_j^i z_j(\tau)$ of inventory of product i residing at A_j during the interval $(\tau - LB_{ij}, \tau]$. Hence, $I_i^i(t)$ no longer represents the total system inventory of product i at time t, but rather the portion of total inventory which is not stock committed by the lag assumptions. (We do not bother to write the inventory balance constraint for $I_j^i(t)$ since it is automatically satisfied by our assumptions.)

To develop a linear programming model, we once again invoke the assumptions of a discrete-time production model with unit-length time periods. If LA_i and $(LT_{ij} + LB_{ij})$ are integers, then all flows in and out of inventory of product i are constant during each interval. Recalling our earlier discussion, inventory balance will be guaranteed for all $t \geq 0$ if it is enforced at the time grid points. Retaining the notation $z_{i\tau}$ for the constant intensity of A_i and u_τ^i for the constant rate of demand for product i during $(\tau - 1, \tau]$, the discrete-time version of (3.11) is

$$I_i^i(t) = I_i^i(0) + \sum_{\tau=1}^{t} z_{i(\tau-LA_i)} - \sum_{j=1}^{M} \sum_{\tau=1}^{t} a_j^i z_{j(\tau+LT_{ij}+LB_{ij})} - \sum_{\tau=1}^{t} u_\tau^i \geq 0, \quad t = 1, 2, \cdots \quad (3.12)$$

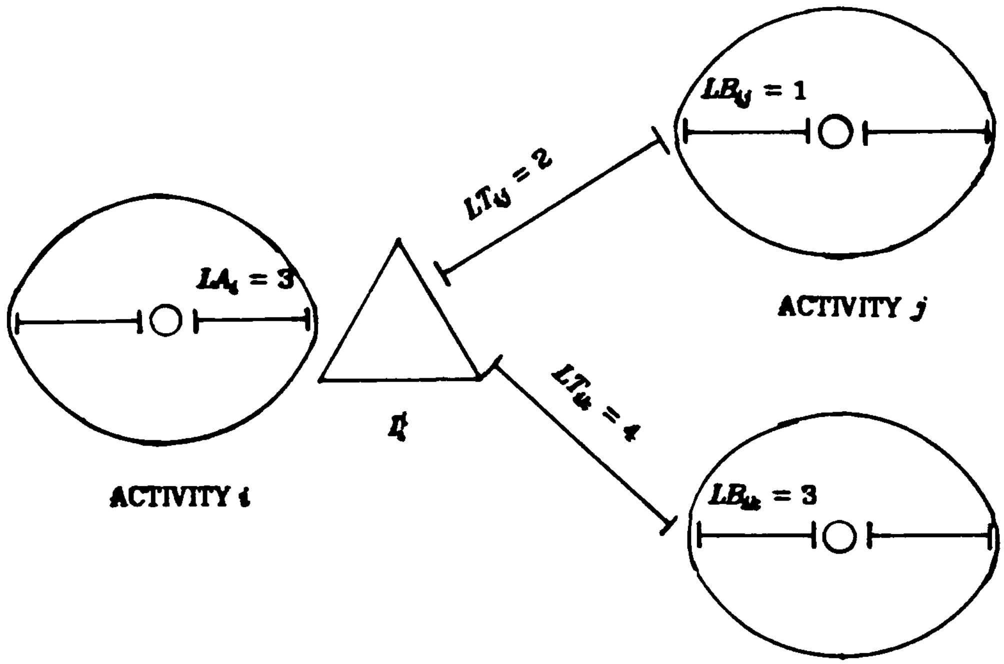
Inspecting the form of (3.11) and (3.12), it appears that separate identification of transfer LT_{ij} and input LB_{ij} lags is unnecessary. This is true under the special assumptions we have made in this model: (1) the only domain constraints on transfers are nonnegativity, and (2) uncommitted inventory of a product is held only where produced. Under alternatives to (1), such as discrete domains for transfers or upper bounds on transfers, it is meaningful to relax (2). In such a case, uncommitted inventory of an intermediate product would be held both where produced and where used, necessitating separate transfer and input lags as well as multiple inventory balance equations to ensure conservation of the product.

Comparison with Familiar Formulations

We contrast (3.12) with familiar formulations incorporating time lags. The inventory balance constraints in the mathematical programming model of Billington *et al.* [1983, 1986] are given in (3.8). Hax and Candea [1985] also use (3.8) without providing any physical definition of "production lead time". Billington *et al.* [1983, 1986] claim their lead time L_i models such phenomena as "time to transfer parts or time for paint to dry." However, any time to transfer parts from intermediate product inventory to an activity which will apply the product as input is a lag of the form LT_{ij}. It is not correct to include allowances for lags of the form LT_{ij} or LB_{ij} in the authors' parameter L_i in (3.8), a point we now elaborate.

Consider the simple example in Figure 1. Application of a quantity of product i at activity j corresponds to resource application at activity i to produce that quantity at least 6 time periods earlier. Similarly, application of product i at activity k corresponds to resource application at activity i at least 10 time periods earlier. If $L_i = 6$, then there are choices for the z_{it}'s and z_{kt}'s *feasible* in (3.8) *but physically impossible.* If $L_i = 10$, then there are choices for the z_{it}'s and z_{jt}'s *physically possible but infeasible* in (3.8). If $6 < L_i < 10$, then both problems arise. On the other hand, there can be no doubt if one uses (3.12) with $LA_i = 3$, $LT_{ij} + LB_{ij} = 3$, $LT_{ik} + LB_{ik} = 7$, since (3.12) properly defines the physically possible choices for the z_{it}'s, z_{jt}'s, and z_{kt}'s.

Even if it were the case that $LT_{ij} + LB_{ij} = LT_{ik} + LB_{ik}$, (3.8) would still be invalid. The issue in this case is precisely what the variables $I^i(t)$ and the parameter $I^i(0)$ in (3.8) represent. Although not explicitly defined in the papers cited above, a common interpretation of $I^i(t)$ in (3.8) is that it represents the "total system quantity" of product i which is "in



LEGEND

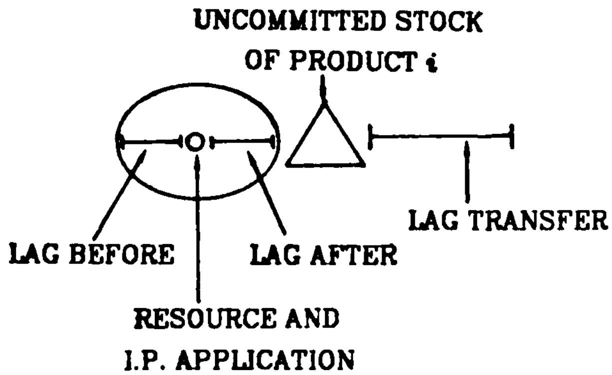


Figure 1

Example Network with Lead Times.

advance of use," i.e., in advance of application at another activity or withdrawal from the system by external demands. From our definition of the elements, this interpretation of $I^i(t)$ is actually given by

$$I^i(t) = I^i(0) + \sum_{\tau=1}^{t} z_i(\tau - LA_i) - \sum_{j=1}^{M}\sum_{\tau=1}^{t} a_j^i z_j(\tau) - \sum_{\tau=1}^{t} u_\tau^i ,$$

which corresponds to (3.8) if one takes $L_i = LA_i$. However, if one includes a positive allowance for $LT_{ij} + LB_{ij}$ in L_i, then $I^i(t)$ in (3.8) no longer represents system inventory of product i at time t, nor for that matter any other measure of inventory of product i at time t. It would be impossible to properly represent initial inventory conditions in (3.8).

In general, any model with transfer or input lags which uses an initial "total system inventory" parameter $I^i(0)$ is flawed: initial conditions are not correctly expressed. That is, $I^i(0)$ will in general include amounts of product i to be immediately applied at a follow-on activity A_j, amounts still being inspected, amounts in transit, and amounts not yet transferred from A_i. When there are shipment or inspection lags, the total quantity $I^i(0)$ should not be simultaneously made available to follow-on activities. On the other hand, in (3.12) the parameter $I_i^i(0)$ properly represents only inventory of product i not yet committed to follow-on activities, while the parameters of the form $z_{j1}, z_{j2}, \ldots, z_{j(LT_{ij}+LB_{ij})}$ serve to specify the status of quantities of product i already sent to activity A_j which are in-transit, being inspected, and to be immediately applied. [1]

---

[1] The correct expression for the total system quantity $I^i(t)$ of product i at time t is derived as follows. $I^i(t)$ is defined by

$$I^i(t) = I_i^i(t) + \sum_{j=1}^{M} [I_j^i(t) + IT_{ij}^i(t)],$$

where $IT_{ij}^i(t)$ denotes the quantity of product i in-transit between A_i and A_j at time t. In the language of the framework,

$$I_j^i(t) = I_j^i(0) + \hat{V}_{ij}^i(t) - Y_j^i(t), \text{ and } IT_{ij}^i(t) = IT_{ij}^i(0) + V_{ij}^i(t) - \hat{V}_{ij}^i(t).$$

Our lag assumptions imply that

$$I_j^i(0) = \int_0^{LB_{ij}} a_j^i z_j(\tau) d\tau \text{ and } IT_j^i(0) = \int_{LB_{ij}}^{LT_{ij}+LB_{ij}} a_j^i z_j(\tau) d\tau .$$

Upon substitution and simplification $I^i(t)$ is expressed as

$$I^i(t) = I_i^i(t) + \sum_{j=1}^{M} [\sum_{\tau=1}^{LT_{ij}+LB_{ij}} a_j^i z_{j(t+\tau)}], \quad t = 1, 2, \cdots$$

Krajewski and Ritzman [1977] propose a "general model" which utilizes balance equations of the form

$$I^i(t) - I^i(t-1) = z_{it} - \sum_{j=1}^{M} a_j^i z_{j(t+L_j)} - u_t^i \qquad (3.13)$$

which is a different special case of (3.11). In this case, L_j cannot reasonably include an allowance for phenomena such as time for paint to dry on product i. Any allowance for $(LA_i + LT_{ij} + LB_{ij})$ included in L_j must be independent of i. If in fact $(LA_i + LT_{ij} + LB_{ij})$ varies with i, then their model incorrectly restricts the intensity of A_j relative to the intensities of its immediate predecessor activities. Unlike the previous special case, this error in the restriction of intensities can be overcome by replacing the authors' parameter L_j in (3.13) with parameters $L_{ij} = (LA_i + LT_{ij} + LB_{ij})$.[2]

However, once again initial conditions can not be expressed correctly. In the model defined by (3.13), product i enters inventory immediately following resource application but before paint is dry. The single parameter $I^i(0)$ in (3.13) can not account for the age distribution of inventory (e.g., dry parts versus those still drying). On the other hand, in (3.12) the parameter $I_i^i(0)$ properly includes only dry parts ready for transfer to follow-on uses, while the parameters $z_{i(1-LA_i)}, z_{i(2-LA_i)}, \ldots, z_{i(0)}$ specify the status of parts which are still drying.

Our final comment of the comparison with familiar formulations concerns the notion of a "frozen planning horizon" (Hax and Candea [1985]). This notion recognizes that production rates over a certain initial period of time must be prespecified in planning calculations. Relative to the measurement of initial inventory at time 0, Billington *et al.* [1983,1986] and Hax and Candea [1985] prespecify production intensities of A_i during the interval $(-L_i, 0]$, while Krajewski and Ritzman [1977] prespecify production intensities of A_j during the interval $(0, L_j]$. The former approach is not correct unless $L_i = LA_i$ and $LT_{ij} + LB_{ij} = 0$ for all j; the latter approach is not correct unless $L_j = LT_{ij} + LB_{ij}$ for all i and $LA_j = 0$. As we have seen, in general, one must prespecify the intensity of A_i during the interval

[2] A similar improvement may be made to MRP calculations. See Section 4.

$$(-LA_i,\ \max_j\{LT_{ji} + LB_{ji}\}],$$

as well as measure $I_i^c(0)$, in order to commence planning calculations at time 0.

A Linear Programming Model With Non-Integer Lead Times

The assumption that lead times are integer is quite restrictive, although it is assumed by all the familiar formulations. We now derive a discrete-time model when lead times are not necessarily integral. We retain the assumption that activity intensities, service capacities, and final demands are constant rates during unit length time periods.

To evaluate inventory balance of product i at time t, one must express the integral functions in (3.11) in terms of the $z_{i\tau}$'s and $z_{j\tau}$'s in a manner which accounts for the non-integer limits of integration. For a real number x, we let x^+ denote the smallest integer greater than or equal to x, and let x^- denote the largest integer less than or equal to x. The first integral in (3.11) is expressed, for all $t \geq 0$, as

$$\int_{-LA_i}^{t-LA_i} z_i(\tau)d\tau = \begin{cases} (t)z_{i(-LA_i)^+} & \text{if } t - LA_i < (-LA_i)^+ \\ [(-LA_i)^+ + LA_i]z_{i(-LA_i)^+} + \sum\limits_{(-LA_i)^+ < \tau \leq (t-LA_i)^-} z_{i\tau} & \text{otherwise.} \\ \quad + [(t - LA_i) - (t - LA_i)^-]\, z_{i(t-LA_i)^+} & \end{cases} \tag{3.14}$$

In (3.14), the coefficients of the first and last terms of the lower expression express the fractions of intensity in the first and last time periods which are included within the limits of integration; the middle term simply sums up intensities of all time periods (if any) in between. The upper expression accounts for the degenerate case when both limits of integration lie in the same time period. Next, let $LX_{ij} \equiv LT_{ij} + LB_{ij}$. The second integral in (3.11) is analogously expressed, for all $t \geq 0$, as

$$\int_{LX_{ij}}^{t+LX_{ij}} z_j(\tau)d\tau = \begin{cases} (t)z_{j(LX_{ij})^+} & \text{if } t + LX_{ij} < (LX_{ij})^+ \\ [LX_{ij}^+ - LX_{ij}]z_{j(LX_{ij})^+} + \sum\limits_{(LX_{ij})^+ < \tau \leq (t+LX_{ij})^-} z_{j\tau} & \text{otherwise.} \\ \quad + [(t + LX_{ij}) - (t + LX_{ij})^-]\, z_{j(t+LX_{ij})^+} & \end{cases} \tag{3.15}$$

Finally, $U^i(t)$ in (3.11) is expressed, for all $t \geq 0$, as

$$U^i(t) = \sum_{\tau=0}^{t} u^i_\tau + (t - t^-) u^i_{t}. \tag{3.16}$$

Substituting the identities (3.14)-(3.16) into (3.11) provides the desired expression of $I^i_t(t)$ in terms of discrete-time variables for any $t \geq 0$. Identifying the flows which are functions of applications or transfers-sent occurring before time 0, we note that for each A_k and for

$$\tau = (-LA_k)^+, (-LA_k)^+ + 1, \ldots, 0, 1, \ldots, \max_j \{LX^+_{jk}\},$$

the intensities $\{z_{k\tau}\}$ must be prespecified.

If the time lags are not integer, ensuring inventory balance (3.11) only at the integer time grid points is not enough to ensure feasibility. Consider the example shown in Figure 2. Here, $U^i = 0$, $LA_i = 1.7$, $LX_{ij} = 1.7$, and $a^i_j = 1$. A simple check shows that (3.11) is satisfied for $t = 1, 2, 3$. However, at $t = 1.7$ cumulative output of i is zero, yet two units are required by j at this time-- a clear infeasibility. This phenomenon can happen since the flows $f^i_i(z_i)$, v_{ij}, and $\hat{v}_{ij}$ are no longer constant on the given time intervals. These flows are still step functions, but the time points where rates of flow may change fall in between the integer time points.

Many investigators overcome this difficulty by either overlooking it and rounding lead times to integer amounts or by subdividing the natural time period so that lead times become integral. The first approach either under-estimates lead time - potentially leading to infeasibilities - or over-estimates lead times - potentially leading to excessive work-in-process. The second approach greatly increases problem size.

To ensure inventory balance (3.11) at all time, we claim it is necessary and sufficient to check (3.11) at all time points when the rates of final demand flows, activity output flows or intermediate product transfer flows may change. For product i let T_i denote this collection of time points. A point in time $t \geq 0$ is an element of T_i if, and only if, $t - LA_i$ or $t + LX_{ij}$ or t itself is an integer. In the example,

$$T_i = \{0, .3, .7, 1.0, 1.3, 1.7, 2.0, 2.3, 2.7, 3.0, \ldots\}.$$

For proof of the claim, note that on the intervals defined by successive points of T_i, all flows

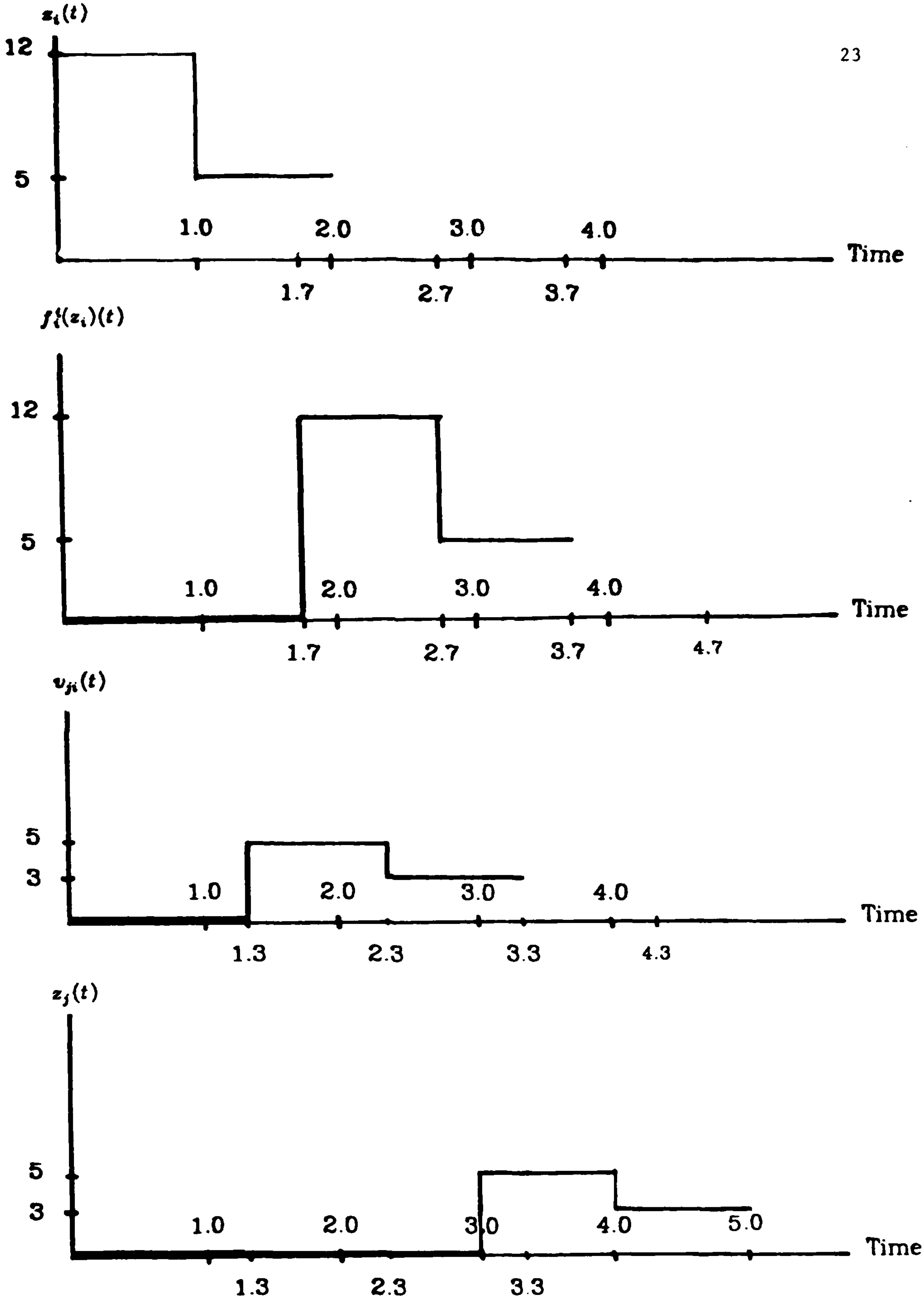


Figure 2

Example of Flows with Non-Integer Lead Times.

in and out of inventory of product i are at constant rates; hence, following our earlier discussion, the inventory level at any time point within such an interval will be nonnegative if the inventory level is enforced to be nonnegative at the end points.

This formulation is not only accurate but it requires significantly fewer constraints than would be included in a model employing a time grid fine enough to make all lead times integer. In the example, our formulation requires 3 balance equations per period for product i. To realize integer lead times a grid with periods of length 0.10 is required for product i, making for 7 extra balance equations within each period. Worse, if one time grid is used to enforce inventory balance for *all* products, an even finer grid might be required. The width of the required grid is precisely the least common divisor of the points in $\bigcup_{i=1}^{M} T_i$. On the other hand, the precision of the lead times does *not* affect the number of balance equations required in our proposed approach.

A Linear Programming Model With Unequal Length Time Periods

Another restrictive assumption of familiar formulations is that all time periods have equal length. However, unequal length periods are desirable, for example, when natural time periods such as weeks include varying numbers of working days because of vacations or holidays. We now modify our previous formulation to handle the case when the lengths of the periods with constant activity intensities are not necessarily equal. Let S denote the set of epochs marking the end points of these periods. To ensure feasibility, it is necessary and sufficient to evaluate (3.11) at those time points t such that $t - LA_i$ or $t + LX_{ij}$ or t is an element of S (thus modifying the definition of T_i). The expressions of the integrals in (3.14), (3.15) and (3.16) under these general conditions must be modified since the coefficients of intensity variables and demand rates were derived assuming a period length of one. To obtain the correct discrete-time version of (3.11), we modify (3.14), (3.15) and (3.16) as follows. First, for a real number x, re-define x^+ (resp. x^-) to mean the smallest (resp. largest) epoch $t_k \in S$ not less than (resp. not greater than) x. Next, substitute the ratios

$$\frac{t}{(-LA_i)^+ - (-LA_i)^-}, \quad \frac{(-LA_i)^+ + LA_i}{(-LA_i)^+ - (-LA_i)^-} \quad \text{and} \quad \frac{(t - LA_i) - (t - LA_i)^-}{(t - LA_i)^+ - (t - LA_i)^-} \tag{3.17}$$

for the coefficients (t), $[(-LA_i)^+ + LA_i]$ and $[(t - LA_i) - (t - LA_i)^-]$ in (3.14), respectively. (We define $\frac{0}{0} \equiv 0$.) The denominators of the ratios in (3.17) express the lengths of the time periods in which the limits of integration lie; the ratios thus express the fraction of intensity in those periods which lies within the limits of integration. Likewise, we substitute the ratios

$$\frac{t}{LX_{ij}^+ - LX_{ij}^-}, \quad \frac{LX_{ij}^+ - LX_{ij}}{LX_{ij}^+ - LX_{ij}^-} \quad \text{and} \quad \frac{(t + LX_{ij}) - (t + LX_{ij})^-}{(t + LX_{ij})^+ - (t + LX_{ij})^-} \tag{3.18}$$

for the coefficients (t), $[LX_{ij}^+ - LX_{ij}]$ and $[(t + LX_{ij}) - (t + LX_{ij})^-]$ in (3.15), respectively. Finally, we modify (3.16) to become, for all $t \geq 0$,

$$U^i(t) = \sum_{\tau=1}^{t} u_\tau^i + \frac{(t - t^-)}{(t^+ - t^-)} u_{t^+}^i . \tag{3.19}$$

Incorporating the modifications (3.17), (3.18) and (3.19), inventory balance constraints (3.11) enforced at the set of epochs T_i for product i, $i = 1, \ldots, M$, and capacity constraints (3.4) enforced for each epoch in S form a more general linear programming production model. The overall time grid for this discrete-time model is $\Lambda = \bigcup_{i=1}^{M} T_i$.

In sum, we have developed an accurate yet parsimonious linear programming model for product-generated networks with three different types of lead times for each product. A time grid (with possibly unequal-length periods) for resource allocation is assumed to be given. Activity input applications are constant on these intervals according to Leontief domains. Based on the lead times, distinct sets of epochs for each product are established for enforcing inventory balance.

Our more general formulation is by no means the most general linear programming production model. The lead times in the foregoing model serve to relax the assumption of instantaneousness of intermediate product applications, outputs, and transfers which is assumed by linear programming models without time lags. The assumption of instantaneous applications of service resources can also be relaxed, i.e., the assumption of Leontief domains is not necessary for obtaining a linear programming model. For example, Krajewski and Ritzman [1977] propose a formulation with resource input coefficients which distribute resource application over an integral number of time periods between intermediate product application and realization of output. Using the framework, we could extend their model to admit noninteger durations for resource applications.

As a final remark, if there are significant processing times (i.e., noninstantaneous service applications), these times also must be accounted for in the lags separating intermediate product input applications and consequent output of each activity. It is not correct to delete processing times from the lags appearing in inventory balance equations, as has been proposed by Billington *et al.* [1986]. Moreover, setup times introduce more breakpoints in inventory input and output flows; if setup times are fractional, enforcing inventory balance only at integer times does not guarantee inventory balance throughout continuous time. A full development of linear programming models for the case of activity production functions with noninstantaneous service applications is beyond the scope of this paper; however, linear programming models for particular noninstantaneous cases of interest are developed in Hackman and Leachman [1985a, 1985b, 1986] and Leachman [1986].

4. MANUFACTURING RESOURCES PLANNING

Manufacturing Resources Planning (MRP) systems convert time-phased final output requirements into requirements for intermediate products and raw materials, and into requirements for service resources. The logic of these conversions is described as a "time-phased explosion" through a "bill of materials" structure (Orlicky [1975]). In this section, we redefine MRP in terms of the framework. MRP is seen as a schedule of intermediate product transfers rather than a schedule of production. We provide a simple analysis of why the use of MRP may result in excessive work-in-process inventories. Alternatives and modifications to standard MRP logic are discussed.

The MRP Model

Consider an acyclic, product-generated, normal, discrete-time production model with unit-length time periods. There are no exogenous materials in the model. (All raw materials of interest are included as products in the network.) The activities (products) are ordered such that if A_i supplies intermediate product to A_j, then $i < j$. Final transfer requirements $v_{i,N+1}$, $i = 1, 2, \ldots, N$, are specified for the system. (Since the model is product-generated, we suppress the superscripts on the flows.) Let $V_i(t) = \sum_{j=i+1}^{N+1} V_{ij}(t)$ denote the cumulative total transfer requirement of A_i at time t. Let $R_i(t) = \max\{0, V_i(t) - I^i(0)\}$. In MRP parlance, $v_i(t)$ is the "gross requirement" for product i at time t, while $r_i(t)$ is the "net requirement" for product i at time t. Final transfer requirements $\{v_{i,N+1}(t)\}$ are termed the "master production schedule". In MRP systems, gross and net requirements, and the master production schedule are expressed as event-based flows on an equal-length discrete time grid such as weeks.

MRP systems include a material requirements planning (mrp) module to determine requirements for products and materials. Coefficients $\{a_j^i\}$ are prespecified, indicating the amount of product i required as input to produce one unit of product j. So-called "lead times" $\{L_i\}$ are also prespecified. Each L_i is an integer which defines a time offset between the event when a transfer requirement for product i is due and the event when intermediate product inputs used to produce the requirement are withdrawn from inventory by A_i. For $i = N, N-1, \ldots, 1$ and for all t, the following computations are performed recursively:

$$V_i(t) = \sum_{l=i+1}^{N+1} V_{il}(t) \tag{4.1}$$

$$R_i(t) = \max\{0, V_i(t) - I^i(0)\} \tag{4.2}$$

$$V_{ji}(t) = a_i^j R_i(t + L_i), \text{ all } j < i. \tag{4.3}$$

An additional step is sometimes included between (4.2) and (4.3) to modify the flow $R_i(t)$ to reflect desirable lot sizes. The (modified) flows $\{r_i\}$ are the result of mrp calculations; each flow r_i defines order quantities of product i due at the time grid points.

MRP systems also include a capacity requirements planning (crp) module to determine "requirements" for service resources. Coefficients $\{a_i^k\}$ are prespecified, indicating the amount of service k required as input to produce one unit of product i, $i = 1, 2, \ldots, N$. Following execution of the mrp module, the crp module estimates the system requirement for service k in period t as

$$\sum_{i=1}^{N} a_i^k r_i(t) \tag{4.4}$$

If (4.4) exceeds capacity of service k in period t, then one is obliged to reduce in some way the final transfer requirements $v_{i,N+1}$, $i = 1, 2, \ldots, N$.

Analysis of the mrp Module

MRP systems incorporate an incomplete model of production. Activity application flows, activity production functions and domain constraints are not defined. In general, the requirements curve R_i for activity A_i is *not* the cumulative output curve $F_i(Y_i)$ of A_i; similarly, the transfer curve V_{ji} is not the cumulative curve Y_i^j of application of product j at A_i. In order for MRP to be valid, fundamental inventory balance at each activity A_i must be ensured, i.e.,

$$F_i(Y_i) \geq R_i \tag{4.5}$$

and

$$Y_i^j \le V_{ji}. \tag{4.6}$$

Instead of specifying Y_i and $F_i(Y_i)$, MRP systems replace (4.5) and (4.6) with the relationship (4.3). Attempting to ensure feasibility solely using (4.3) can lead to excessive work-in-process inventory, as we now discuss.

For the purposes of illustration, we consider application of MRP to a production system satisfying the assumptions of Section 3.2. We further assume that the rate of production (per period) by activity A_i is bounded by $\bar{z}_i$. For example, $\bar{z}_i$ may reflect an allocation of service resources to A_i. Plotted in Figure 3 is a particular net requirements curve R_i. Also plotted is the "latest" output curve Z_i^L satisfying the requirements R_i. Z_i^L is generated by assuming it is feasible to produce up to $\bar{z}_i$ in each period. We plot the case in which $r_i(t)$ exceeds $\bar{z}_i$ in various periods, as is frequently the case in actual practice.

Suppose one were to produce item i according to the "late" schedule dictated by the output curve Z_i^L shown in Figure 3. Because the intermediate products necessary for production *must be available prior to production, if item j is required in production of item i,* then $V_{ji} \ge Y_i^j = a_i^j \cdot Z_i^L$. To minimize the inventory of item j at A_i one would take $V_{ji} = a_i^j \cdot Z_i^L$. Note that one must calculate Z_i^L from R_i in order to obtain the V_{ji} which minimizes inventory. (A simple algorithm for calculating Z_i^L from R_i in a discrete-time model is given in Leachman [1979].)

However, in an MRP system, V_{ji} is calculated using (4.3). Ideally, the time shift used should be the smallest such shift guaranteeing that

$$V_{ji}(t) \ge a_i^j Z_i^L(t). \tag{4.7}$$

In this example, $L_i = 4$ is the minimal such time shift. (See Figure 3.) In general, the smallest L_i is dependent on the particular net transfer requirements curve R_i, and the bound on the rate of production $\bar{z}_i$. As discussed before, L_i is exogenously prespecified, *independent* of R_i and $\bar{z}_i$. A large L_i must be chosen to ensure feasibility. Even if L_i were the minimum shift satisfying (4.7), the insistence on a simple time shift model as expressed in (4.3) in general

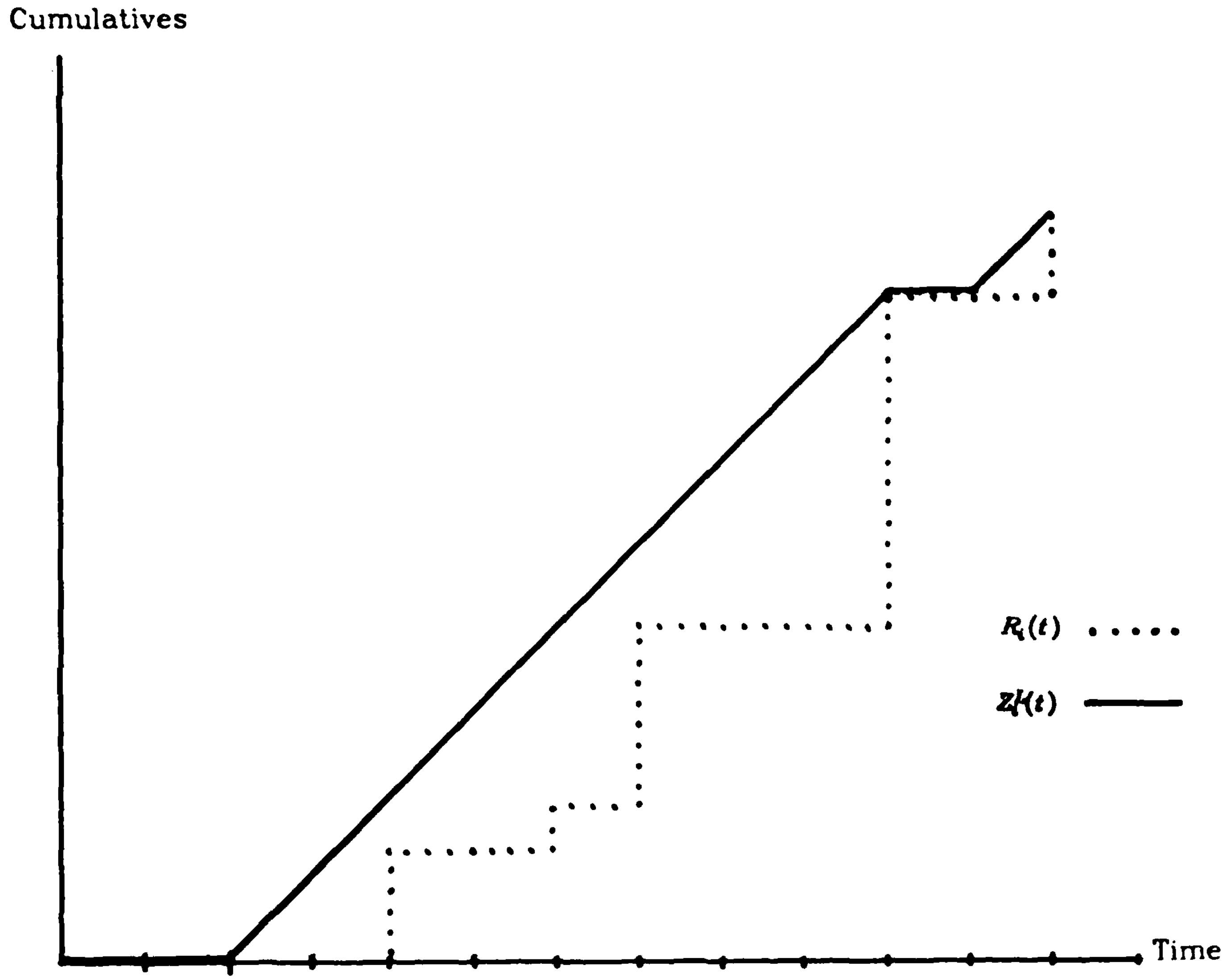


Figure 3

Example of Net Requirements and "Latest" Intensity Curves.

results in excessive intermediate product inventories, as depicted in Figure 4.

In Figure 4, $a_i^j = 1$, and the cumulative transfer of product j to A_i is taken to be $R_i(t + 4)$ (i.e., a time shift of required output). Suppose production of item i proceeds at the maximum rate allowable by the bound on the production rate and the rate of intermediate product j transferred to it. The resulting cumulative output curve is labeled Z_i^E (the "early curve"). By following Z_i^E instead of Z_i^L, the inventory of intermediate product j is significantly reduced. However, the inventory of item i awaiting transfer becomes significant. Actual production must lie somewhere in between the Z_i^E and Z_i^L curves. Hence, when a simple time shift model is used to relate transfers in lieu of production functions, inventory of product awaiting production or inventory of product awaiting transfer will in general be large.

To reduce excess inventory, one must calculate output curves from requirements curves, and then calculate transfers from output curves. If service resources are preallocated to activities (as in the example), these calculations are easy to incorporate into the mrp calculation (Leachman [1979]). Alternatively, one can try to solve the mathematical programming problem representing simultaneous resource allocation and scheduling (Billington *et al.* [1983, 1986]).

As discussed in Billington *et al.* [1983, 1986], the programming formulation is computationally intractable for realistic problems. The mrp calculation remains a popular method for large-scale scheduling; moreover, inflation of the lead time parameters is a convenient means of addressing stochastic aspects of production which are difficult to model. Nonetheless, a simple improvement to the mrp calculation can be made which is analogous to the improvement we suggest for l.p. formulations incorporating lags. Consider application of MRP to a production system satisfying the assumptions of Section 3.3. The mrp parameter L_i must account for time to transfer inputs from predecessors (LT_{ji}), time to inspect inputs (LB_{ji}), time to produce (LA_i), as well as time spent in inventory I_i^j (i.e., time resulting from differences between output and requirements curves, as discussed above). The first two fixed lags depend on the source of inputs; hence the mrp parameter L_i in (4.3) must include an

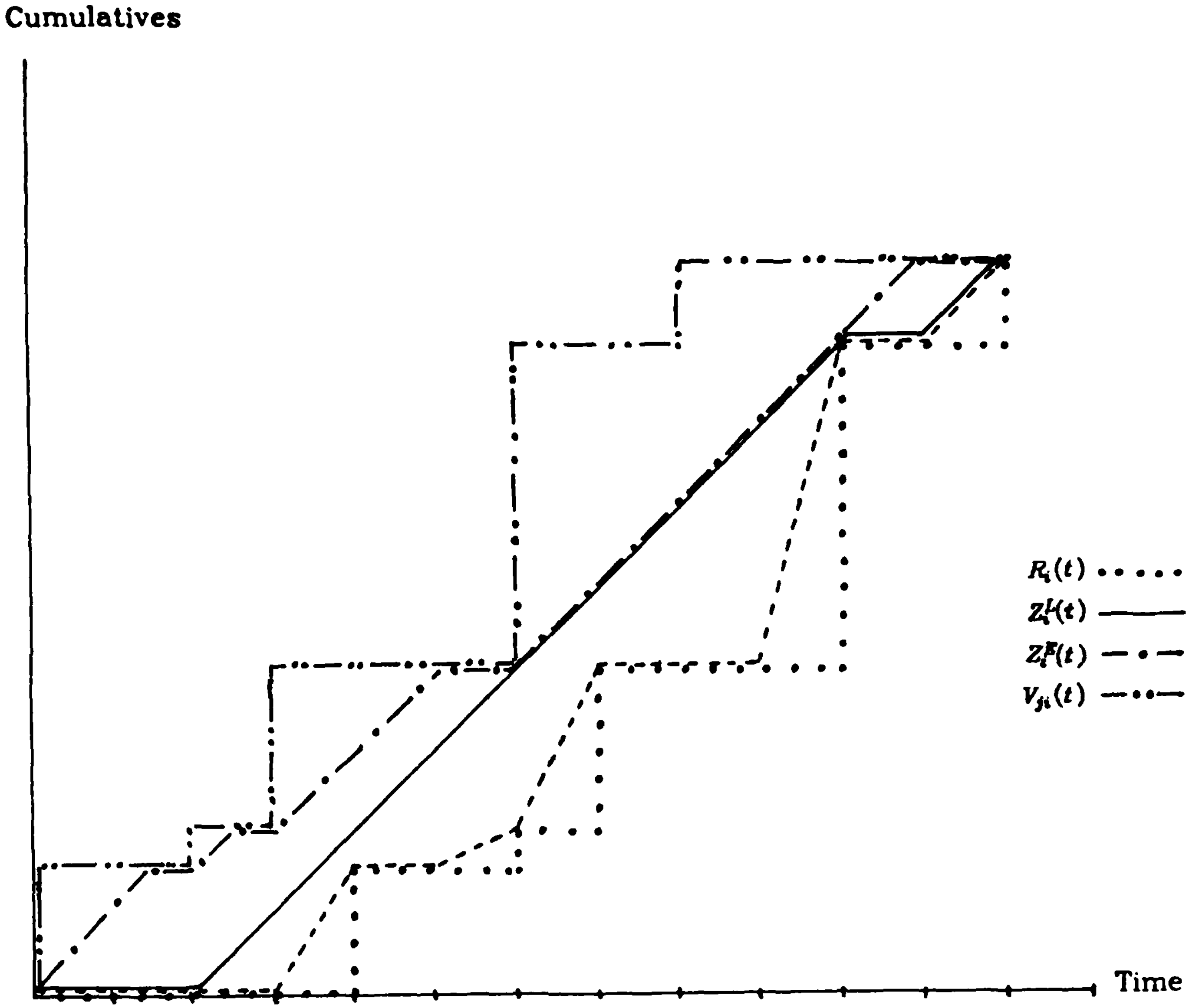


Figure 4

Example of Net Requirements, Transfer, Latest and Earliest Intensity Curves.

allowance for $\max_r(LT_{ri} + LB_{ri})$ in addition to other factors. In the case of multiple activity inputs with different lags, some reduction in lead time (and excess inventory) is obtained by substituting for L_i in (4.3) a parameter L_{ji} which is calculated the same as L_i except that the allowance above is replaced by $(LT_{ji} + LB_{ji})$.

Analysis of the crp Module

We now turn to the representation of service resources in MRP. From the point of view of the framework it is not valid to estimate service resource loads in terms of transfer requirements, as is done by the crp module. As we have seen, even when activity production functions are Leontief the requirements curve for an activity A_i is not its intensity curve.

An infeasibility suggested by the crp module does not necessarily imply that the final transfer requirements are infeasible, nor even that the MRP-derived transfer schedules are infeasible. Consider once again our example in Figure 4. The dashed curve is the continuous flow corresponding to the event-based $R_i(t)$ curve. The crp module estimates resource consumption by A_i in terms of the dashed curve. Since its slope exceeds $\bar{z}_i$ (the slope of $Z_i^L(t)$) in several periods, the crp module would indicate that $R_i(t)$ is infeasible; however, Figure 4 clearly shows that $R_i(t)$ is feasible with respect to $\bar{z}_i$.

We have seen that MRP systems incorporate a model of production in which all flows are expressed in terms of intermediate product transfers. On the other hand, basic linear programming models incorporate a model of production in which all flows are expressed in terms of intensities of activity input application. In a more accurate model of multi-stage production, it is likely that both application and transfer flows must be explicit in the model in order to represent the various domain constraints peculiar to each type of flow.

5. CRITICAL PATH MODELS

In familiar critical path models (CPM), an acyclic network of activities $A_1, \ldots, A_N$ is given. Each A_i is assumed to operate in some un-interrupted interval of time with integer length $d_i > 0$. In the activity-on-node format, an arc from A_i to A_j means there is a *strict precedence* relationship between A_i and A_j. (A_j can not be started until after A_i has finished.) In the standard application of resource-constrained CPM, resource use by an activity is assumed constant during its duration (Moder et al. [1985]).

Resource-constrained CPM embeds a model of production as follows. The activity network is the CPM activity-on-node network, exclusive of the source for exogenous inputs and the sink for final outputs. The underlying production system has the following characteristics.

(1) The production model is acyclic and normal with a finite horizon.

(2) Each activity produces a vector of outputs, whereby a distinct intermediate product is supplied to each follower. Activities with no followers produce a final product transferred to a sink node. Since we may identify uniquely the products of the system with arcs in the network, we replace the notation m for product with the notation (i,j).

(3) Exactly one unit of each product is produced.

(4) Transfers $v_{ij}^{(i,j)}$ are event-based flows. Quantity transferred equals 0 or 1. We take the time grid Λ to be the set of nonnegative integers.

(5) Activity production functions have a restricted Leontief domain. Input applications to A_i are indexed by a "box" curve of the form

$$z_i(\tau) = \begin{cases} \dfrac{1}{d_i} & \text{if } \tau \in (S_i, S_i + d_i] \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

where $S_i \in \Lambda$ corresponds to the start time of A_i. Domain constraints on allocations are assumed to be trivial.

(6) Cumulative output of an activity is measured in terms of the fraction of required

resources which have been applied. We define $f_i^{(i,j)}(z_i) = z_i$ for each j such that (i,j) corresponds to an arc in the network. Note that an activity *simultaneously* produces one product for each follow-on activity.

It is a simple matter to verify that strict precedence is implied by the fundamental inventory balance equation and the assumptions (1)-(6). Hence strict precedence between activities is simply a particular form of inventory balance which arises when there are event-based transfers.

The representation (1)-(6) of CPM in terms of the framework changes according to one's conventions for describing the physical phenomena. Assumption (6) is based on a particular convention for output measurement, e.g., when 50% of the required effort has been made, an activity is considered 50% done. A second representation arises if activity output is viewed as a discrete "lump" emerging exactly when resource applications are complete. In this representation, we relax the integer constraint on $v_{ij}^{(i,j)}$ in (4) and replace (6) by

$$f_i^{(i,j)}(z_i)(\tau) = \begin{cases} 1 & \text{if } \tau = S_i + d_i \text{ and } (i,j) \text{ is an arc} \\ 0 & \text{otherwise.} \end{cases} \tag{5.2}$$

Now, each production function has Leontief domain but is no longer Leontief.

A third representation arises if instead of the output or transfers being discrete, the follow-on applications of intermediate product are viewed as discrete. In this case, we maintain the original assumption (6), relax the domain for transfers in (4) to simple nonnegativity, and alter (5) such that

$$y_j^{(i,j)}(\tau) = \begin{cases} 1 & \text{if } \tau = S_j \text{ and } (i,j) \text{ is an arc} \\ 0 & \text{otherwise.} \end{cases} \tag{5.3}$$

Now, the domains for the production functions are no longer Leontief; only service resource applications (y_i^k) are indexed by an intensity function (5.1).

In fact, a strict precedence model may be formulated *without* event-based flows. In this fourth model, there is no inventory of product (i,j) at A_i, i.e., (4) becomes $v_{ij}^{(i,j)} = f_i^{(i,j)}(y_i)$. Each production function has restricted Leontief domain exactly as in (5). With these assumptions, we can maintain strict precedence if we replace (6) by

$$f_i^{(i,j)}(z_i)(\tau) = \begin{cases} \dfrac{1}{d_j} & \text{if } \tau \in (S_i + d_i, S_i + d_i + d_j] \text{ and } (i,j) \text{ is an arc} \\ 0 & \text{otherwise.} \end{cases} \tag{5.4}$$

The production function defined in (5.4) maps the intensity curve for A_i into the *earliest* intensity curve for A_j consistent with the assumption of strict precedence. That is, the production function determines the appropriate *bound* on the choices for intensity of A_j. This last representation is the only one of the four which is preserved under aggregation of resource-constrained CPM networks; using this model, a computationally tractable approach to multi-project aggregate planning has been developed (Hackman and Leachman [1985a, 1985b], Leachman and Boysen [1985]). See Figure 5 for a pictorial representation of the four alternative models.

Comparing CPM to the basic l.p. model in Section 3.2, note that both models utilize the same production function, except that application flows in CPM have the severely restricted domain defined by (5.1). It is natural to consider relaxation of these domain constraints to allow flexibility in activity operation akin to that allowed by the activity analysis model in linear programs. In many project-oriented physical systems, the application of service resources need not be at one fixed rate from start to finish of project activities. (See, for example, the discussion of the application of trade labor services to ship overhaul activities in Leachman [1979] and Leachman [1983].)

A number of authors have developed extensions in this regard, which we now categorize in terms of the allowed domains for application flows. Wiest [1967] and Talbot [1982] have considered resource-constrained scheduling when there are discrete alternatives for the duration (and associated resource requirements) of each activity. However, from the point of view of the framework, each alternative still corresponds to a "box" intensity curve for each activity, i.e., a constant rate of resource application from start to finish of an activity. Leachman [1983] develops a technique for resource leveling when there is a continuous range of

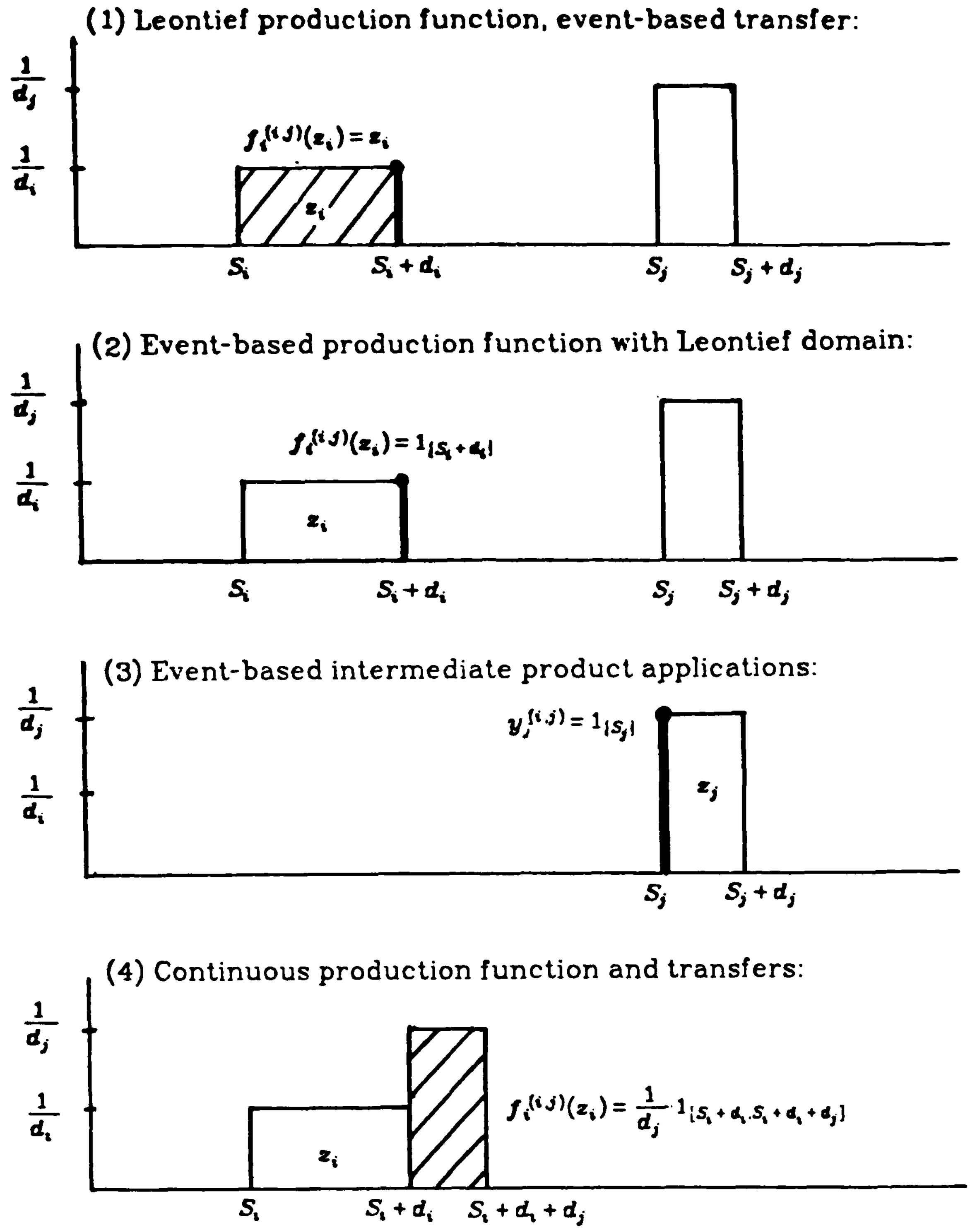


Figure 5

Alternative Representations of CPM.

possible intensity levels for each activity, i.e., (5.1) is replaced by

$$z_i(\tau) = \begin{cases} \dfrac{1}{d_i} & \text{if } \tau \in (S_i, S_i + d_i] \text{ for some } S_i \in R_+ \text{ and } \dfrac{1}{d_i} \in [\underline{z}_i, \bar{z}_i] \\ 0 & \text{otherwise.} \end{cases} \tag{5.5}$$

Weglarz [1981] and Dincerler [1984] relax this domain even further in considering resource-constrained scheduling. They develop scheduling algorithms which exploit the assumption that intensity may vary during the duration of an activity, i.e., (5.1) is replaced by

$$\{z_i\text{: for some } S_i,\ F_i \in R_+,\ z_i \text{ is nonzero only on the interval } [S_i, F_i], \text{ and for } \tau \in [S_i, F_i],\ \underline{z}_i \le z_i(\tau) \le \bar{z}_i\}. \tag{5.6}$$

In summary, we have cast CPM as a model of production, recognizing that each arc in the CPM network represents the transfer of a distinct intermediate product, and that a strict precedence relationship is an expression of inventory balance of such a product. Our simple categorization of extensions to CPM in terms of relaxations of the domain for applications illustrates the power of the framework to compare models easily yet rigorously.

6. A STRUCTURED APPROACH TO MODELING PRODUCTION

We have introduced a general framework for deterministic models of production and illustrated its use through the evaluation and improvement of familiar models. However, the chief intent of our framework is to guide the formulation of specific models of production. By "specific" we mean that a particular physical production process is considered, the planning/scheduling decision levels are specified, and one seeks to construct a model of production for each level.

The process of model-building using the framework proceeds in an orderly, structured fashion as follows:

Step 1: Develop a detailed model of production.

(a) Elucidate assumptions about the physical system.

(b) Identify the model elements (production functions, domain constraints, etc.).

(c) Express constraints which ensure conservation through continuous time of all services, materials, and products of interest.

(d) Approximate as necessary.

Step 2: Develop an aggregate model of production for the decision level under consideration (if required).

Repeat steps (a)-(d) above for the aggregate level of detail.

Step 3: Test and validate the model.

Note that we develop an aggregate model appropriate for the decision requirements of the planning problem having *first* developed an accurate model of the production system. We view aggregation as a meta-model operation rather than a model-specific operation. In our framework, aggregation requires extensive *modeling* effort. It leads from one (detailed) model consistent with the general framework to another (aggregate) model consistent with the framework. In our experience, these two models may - and often do - have dramatically different structure. See, for example, Leachman and Boysen [1985] and Hackman and Leachman [1985a, 1985b]. On the other hand, previous treatments of aggregation consider only model-specific aggregation which is a largely mathematical operation for reducing the dimension of a model while preserving its structure. It leads from one instance of a model (the detailed-level) to another instance (the aggregate-level) having the same structure. For example, row and column aggregation begins with a linear program whose primitive elements (A, b, c, and x) are already specified. It leads - without additional modeling - to another, albeit smaller, linear program. (We do not mean to say that model-specific aggregation techniques are not valuable. In fact, they may be useful components of solution techniques for aggregate models developed via our structured approach.)

In this paper we have illustrated only Step 1 of the structured approach, by way of analysis of the familiar models. We derived linear programming, MRP and CPM models using the framework from explicit assumptions about the physical system. Examples of the development of novel aggregate production models using the structured approach and the framework may be found in Hackman and Leachman [1985a, 1985b, 1986].

ACKNOWLEDGEMENTS

This research was supported in part by the Office of Naval Research and the Puget Sound Naval Shipyard under Contract N00014-76-C-0134 with the University of California and by a gift to the University of California from Intel Corporation. The authors would especially like to thank John H. Vande Vate, John A. Harrison, and Renato Monteiro for insightful comments on earlier drafts. Encouragement from our colleagues John J. Bartholdi, Loren K. Platzman, Craig A. Tovey, and Betsy S. Greenberg is also gratefully appreciated.

REFERENCES

ABRAHAM, C., B. DIETRICH, S. GRAVES, W. MAXWELL, and C. YANO. 1985. "Research Agenda for Models to Plan and Schedule Manufacturing Systems," presentation at TIMS/ORSA Joint National Meeting, Boston, MA., May 4, 1985.

AKINC, A., and G. M. ROODMAN. 1986. "A New Approach to Aggregate Production Planning," *IIE Transactions*, **18** (1), 88-94.

BENSOUSSAN, A., M. CROUHY and J. M. PROTH. 1985. *Mathematical Theory of Production Planning*, North Holland, New York.

BILLINGTON, P. J., J. O. MCCLAIN and L. J. THOMAS. 1983. "Mathematical Programming Approaches to Capacity-Constrained MRP Systems: Review, Formulation and Problem Reduction," *Management Science*, **29** (10), 1126-1141.

BILLINGTON, P. J., J. O. MCCLAIN and L. J. THOMAS. 1986. "Heuristics for Multilevel Lot-Sizing With a Bottleneck," *Management Science*, **32** (8), 989-1006.

DEMPSTER, M. A. H., M. L. FISHER, L. JANSEN, B. J. LAGEWEG, J. K. LENSTRA and A. H. G. RINNOOY KAN. 1981. "Analytical Evaluation of Hierarchical Planning Systems," *Operations Research*, **29** (4), 707-716.

DINCERLER, A. 1984. "Project Scheduling in Project-Oriented Production Systems," unpublished D. Eng. Dissertation, Dept. of Industrial Engineering and Operations Research, University of California, Berkeley.

GEOFFRION, A. M. 1985. "Structured Modeling," Graduate School of Management Research Report, University of California, Los Angeles.

HACKMAN, S. T. 1984. "A General Model of Production: Theory and Application," ISyE Report C-84-2, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta.

HACKMAN, S. T. and R. C. LEACHMAN. 1985a. "An Aggregate Model of Project-Oriented Production. I: Construction of a Dynamic Production Function Based on a General Framework," ISyE Report J-85-8, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta.

HACKMAN, S. T. and R. C. LEACHMAN. 1985b. "An Aggregate Model of Project-Oriented Production. II: Replications and Aggregations of the Primitive Network Structure," ISyE Report J-85-9, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta.

HACKMAN, S. T. and R. C. LEACHMAN. 1986. "Aggregate Models of Production," forthcoming ORC Report, Operations Research Center, University of California, Berkeley.

HAX, A. C. and D. CANDEA. 1985. *Production and Inventory Management*, Prentice-Hall, Englewood Cliffs, N.J.

JOHNSON, L. A. and D. C. MONTGOMERY. 1974. *Operations Research in Production Planning, Scheduling and Inventory Control*, John Wiley and Sons, New York.

KRAJEWSKI, L. and L. RITZMAN. 1977. "Disaggregation in Manufacturing and Service Organizations: Survey of Problems and Research," *Decision Sciences*, **8** (1).

LEACHMAN, R. C. 1979. "Aspects of Dynamic Production Planning," ORC Report 79-8, Operations Research Center, University of California, Berkeley.

LEACHMAN, R. C. 1983. "Multiple Resource Leveling in Construction Systems Through Variation of Activity Intensities," *Naval Research Logistics Quarterly*, **30** (3), 187-198.

LEACHMAN, R. C. 1986. "Preliminary Design and Development of a Corporate-Level Production Planning System for the Semiconductor Industry," ORC 86-11, Operations Research Center, University of California, Berkeley.

LEACHMAN, R. C. and J. BOYSEN. 1985. "An Aggregate Model for Multi-Project Resource Allocation," in *Project Management: Methods and Studies*, B. V. Dean (ed.), North Holland, New York.

LEONTIEF, W. W. 1951. *The Structure of the American Economy*, Oxford University Press, Oxford, England.

MODER, J. J., C. R. PHILLIPS and E. W. DAVIS. 1985. *Project Managemant with CPM and PERT*, 3rd Edition, Van Nostrand Reinhold, New York.

ORLICKY, J. 1975. *Material Requirements Planning*, McGraw-Hill, New York.

SHEPHARD, R. W. 1970. *Theory of Cost and Production Functions*, Princeton University Press, Princeton, N.J.

SHEPHARD, R. W., R. AL-AYAT and R. C. LEACHMAN. 1977. "Shipbuilding Production Function: An Example of a Dynamic Production Function," in *Quantitative Wirtschaftsforschung Festschrift Volume (Wilhelm Krelle)*, H. Albach (ed.), J. B. C. Mohr, Tubingen, W. Germany.

TALBOT, F. B. 1982. "Resource-Constrained Project Scheduling with Time-Resource Tradeoffs: The Nonpreemptive Case, *Management Science*, **28** (10), 1197-1210.

WEGLARZ, J. 1981. "Project Scheduling with Continuously-Divisible, Doubly-Constrained Resources," *Management Science*, **27** (9), 1040-1053.

WIEST, J. D. 1967. "A Heuristic Model for Scheduling Large Projects with Limited Resources," *Management Science*, **13** (6).

END

2-87

DTIC